**COMANDOS ÚTEIS PARA UTILIZAÇÃO EM PYTHON**

Organizar a ideia melhor sobre como colocar esses comandos

1. **Sumário**

[1. Métodos Comuns para Strings 2](#__RefHeading___Toc25_1549371939)

[1.1. Método len(): 2](#__RefHeading___Toc35_1549371939)

[1.2. Métodos upper() e lower(): 2](#__RefHeading___Toc37_1549371939)

[1.3. Método isalpha(): 2](#__RefHeading___Toc39_1549371939)

[1.4. Método isnumeric(): 3](#__RefHeading___Toc62_1549371939)

[1.5. Método strip(): 3](#__RefHeading___Toc41_1549371939)

[1.6. Método join(): 3](#__RefHeading___Toc64_1549371939)

[1.7. Métodos Split 4](#__RefHeading___Toc27_1549371939)

[1.8. Método capitalize(): 4](#__RefHeading___Toc66_1549371939)

[1.9. Método count(): 4](#__RefHeading___Toc68_1549371939)

[1.10. Método replace(): 5](#__RefHeading___Toc70_1549371939)

[1.11. Método find(): 5](#__RefHeading___Toc72_1549371939)

[1.12. Método title(): 5](#__RefHeading___Toc74_1549371939)

[2. Alinhamento de F-String: 6](#__RefHeading___Toc76_1549371939)

[2.1. F-string e suas formatações: 6](#__RefHeading___Toc285_1934664425)

[2.2. Alinhamento a Esquerda, Direita e Centro: 7](#__RefHeading___Toc78_1549371939)

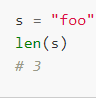
[Referências 8](#__RefHeading___Toc18_1549371939)

# Métodos Comuns para Strings

## Método len():

O comando len() vai servir para mostrar o tamanho da string. Ele é uma função interna do Python e retorna o comprimento de um objeto, por exemplo, ele pode retornar o número de itens em uma lista. Pode-se usar a função com muitos tipos de dados diferentes.

**Exemplo**

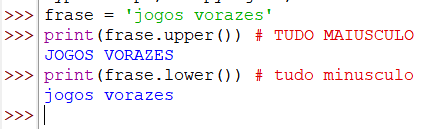
****

## Métodos upper() e lower():

O método upper() recebe o valor de uma string e retorna ela mesma, mas com todos os caracteres em maiúscula.

E o método lower é justamente o contrário, ele recebe o valor da string e retorna ela mesma com todos os caracteres em minúsculas.

**Exemplo**

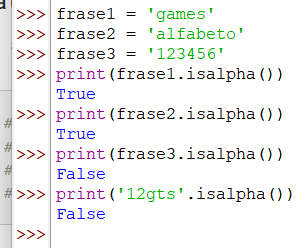
****

## Método isalpha():

O método isalpha retorna “True” se todos os caracteres na string forem alfabetos, caso contrário, retorna “False”. Esta função é usada para verificar se o argumento inclui apenas caracteres do alfabeto.

Esse método é bastante utilizado para saber se o que foi digitado (em string/str) foi um número ou não, e se retorna False, ele converte para int.

**Exemplo**:

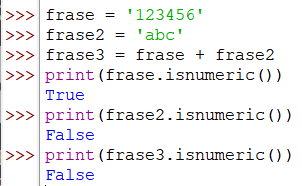


## Método isnumeric():

Isnumeric é um método embutido usado para a manipulação de strings, onde o método retorna “True” se todos os caracteres da string forem numéricos, caso contrário, retorna “False”.

Essa função é usada para verificar se o argumento contém todos os caracteres numéricos, como: inteiros, fações, subscrito, sobrescrito, numerais romanos etc.

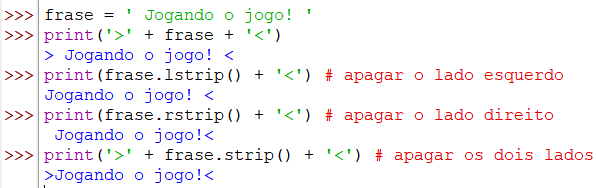
**Exemplo:**

****

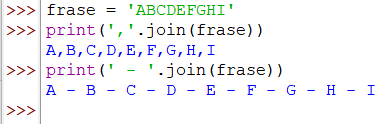
## Método strip():

Python é capaz de encontrar espaços em branco dos lados direito e esquerdo de uma string. Strip pega uma cópia da string que tiver esses espaços em branco do início e no final e retorna sem esses espaços.  
Também é possível apagar apenas um dos lados (esquerdo ou direito) utilizando o lstrip() - lado esquerdo – e o rstrip() - lado direito

**Exemplo:**



## Método join():

O método join é usado para especificar os elementos de uma sequência de caracteres para gerar uma nova sequência de conexão. Ou seja, junta cada item da string com um delimitador especificado. É o inverso do split() e também é aceito em listas.

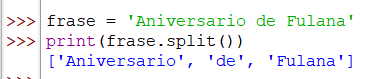
**Exemplo:**

## Métodos Split

O método split é uma das funções disponíveis em Python utilizada para a manipulação de strings. Na prática, ele permite dividir o conteúdo da variável de acordo com as condições especificadas em cada parâmetro da função ou com os valores predefinidos por padrão.

O método split() retorna uma lista de strings após quebrar a string dada pelo separador especificado.

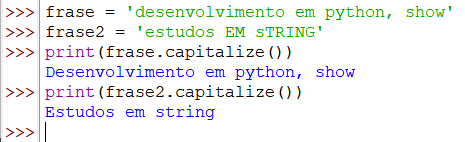
**Exemplo:**

****

## Método capitalize():

O método retorna uma cópia da string original e converte o primeiro caractere da string em uma letra maiúscula enquanto transforma todos os outros caracteres na string em letras minúsculas.

**Exemplo:**

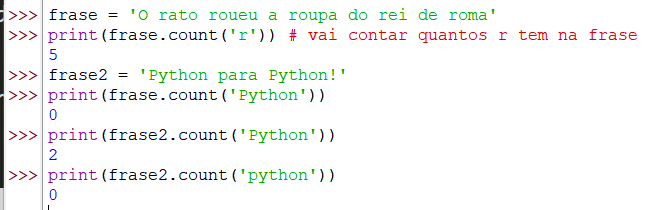
****

## Método count():

Count é uma função embutida na linguagem de programação Python que retorna o número de ocorrências de uma substring na string fornecida. O método count() retorna um inteiro que denota o número de vezes que uma substring ocorre em uma determinada string.

Também a diferenciação de letras maiúscula e minúsculas.

**Exemplo:**

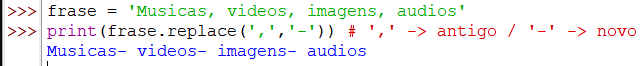
****

## Método replace():

O método replace(), recebe dois argumentos, para fazer a localização e substituir valores dentro de uma string.

Na sua sintaxe, primeiro vem o argumento antigo e logo depois o novo.

**Exemplo:**

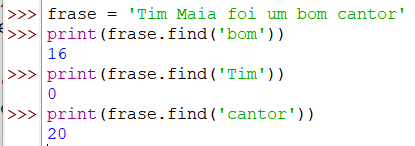
****

## Método find():

O método find deve ser utilizado apenas se te interessar a posição da ocorrência na string, isto é, saber em que parte da string foi encontrado o valor desejado. Se a intenção é apenas verificar se for encontrado, o ideal é utilizar o operador in.

Usando o find ele vai retornar o menor índice da primeira aparição dessa substring. Se não encontrar retorna -1.

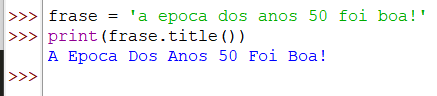
**Exemplo:**

****

## Método title():

A função title é usada para converter o primeiro caractere em cada palavra em maiúsculas e os caracteres restantes em minúsculas na string e retorna uma nova string.

**Exemplo:**

****

# Alinhamento de F-String:

Alinhamento de texto em Python é útil para imprimir saídas formatas e limpas. Algumas vezes, os dados a serem impressos variam em comprimento, o que os torna confusos quando impressos.

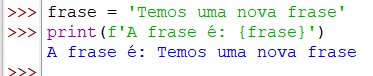
Usando o alinhamento de f-string, a saída pode ser alinhada definindo o alinhamento como esquerdo, direito ou centro e também definindo o espaço (largura) a ser reservado para a string.

Mas antes de continuarmos sobre alinhamento, vamos falar sobre o que é F-string e suas formatações.

## F-string e suas formatações:

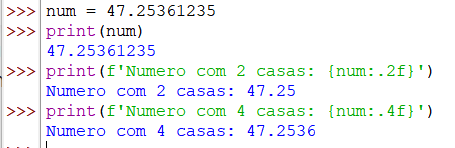
A partir das versões mais recentes do Python, começou-se a usar uma ferramenta para formatação de textos, só que ela não serve apenas e exclusivamente para formatação de textos, ela da mais liberdade para se modificar uma string.

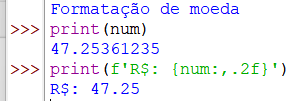
As f-strings vão servir para que você consiga colocar uma variável dentro de um texto, e isso é feito utilizando a letra “f” antes do texto e colocando a sua variável dentro de {} (chaves).

**Exemplo**:

Com isso também nós facilita muito a formatação de outras informações, deixando bem mais fácil nossa vida de programador.

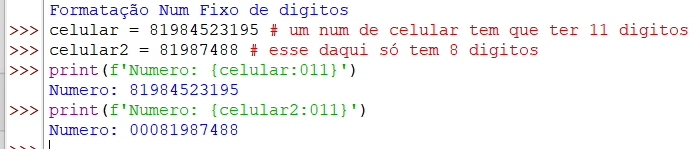
**Exemplo de formatação com casas decimais:**

****

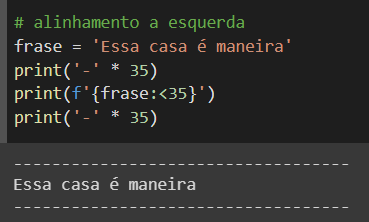
****

**Exemplo de formatação de Moeda:**

**Exemplo de Número fixo de dígitos:**

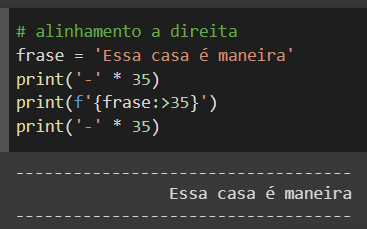
****

## Alinhamento a Esquerda, Direita e Centro:

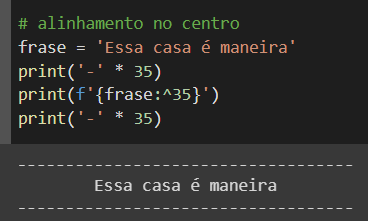


Para a sintaxe da string de saída do alinhamento à **esquerda**, defina ‘<’ seguido pelo número da largura.

**Exemplo:**

Para a sintaxe da string de saída de alinhamento à **direita**, defina ‘>’ seguido pelo número da largura.

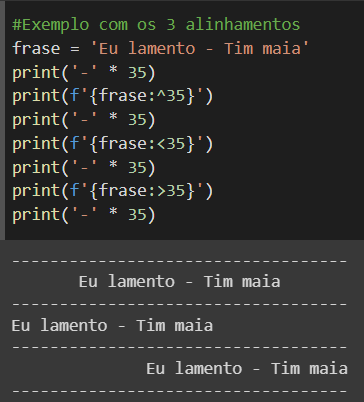
**Exemplo:**

****

Para a sintaxe da string de saída do alinhamento **central**, defina ‘^’ seguido pelo número da largura.

**Exemplo:**

**Exemplo com todos os 3:**

****

# Referências

* **Métodos comuns para strings→**
  + <http://devfuria.com.br/python/strings/>
  + <https://blog.betrybe.com/python/python-split/>
  + <https://www.acervolima.com.br/2021/01/python-string-split.html>
  + https://www.pythonprogressivo.net/2018/10/String-Maiuscula-Minuscula.html
  + http://ptcomputador.com/P/python-programming/93783.html
  + https://acervolima.com/string-capitalize-em-python/
  + https://acervolima.com/metodo-python-string-count-1/
  + https://pt.stackoverflow.com/questions/249532/como-manipular-strings-com-find
* **Alinhamento de F-Strings →**
  + [https://www.hashtagtreinamentos.com/f-strings-em-python#:~:text=As%20f%2Dstrings%20v%C3%A3o%20servir,vari%C3%A1vel%20dentro%20de%20%7B%7D%20chaves.&text=D%C3%A1%20s%C3%B3%20uma%20olhada%20como,informa%C3%A7%C3%B5es%20dentro%20de%20um%20texto](https://www.hashtagtreinamentos.com/f-strings-em-python" \l ":~:text=As f-strings vão servir,variável dentro de {} chaves.&text=Dá só uma olhada como,informações dentro de um texto).
  + https://acervolima.com/alinhamento-de-string-em-f-string-python/